UNIVERSITI SAINS MALAYSIA

First Semester Examination
Academic Session 1997/98

September 1997

CSI502 - Problem Solving and Programming

Duration : [3 hours]

INSTRUCTION TO CANDIDATE:

- Please ensure that this examination paper contains **FOUR** questions in **SIX** printed pages before you start the examination.
- Answer **ALL** questions. You can choose to answer either in Bahasa Malaysia or English.
- Use C programming Language whenever needed.

ENGLISH VERSION OF THE QUESTION PAPER

1. (a) (i) Briefly explains the 4 important phases in problem solving using computer.

(ii) Give two principles for acheiving high cohesion in program design. Why high cohesion is important?

[30/100]

(b) You are requested to design a program which will do the following tasks:

The program will receive three numbers and it will sort them in the following manner: the smallest will be stored in NUM1, the second largest is stored in NUM2, and the largest is stored in NUM3. Finally the program will display the numbers in the order sorted from smallest to largest.

(i) Identify the input/output for the above problem.

(ii) Specify the main modules that you will need to solve the problem. Each module must work on specific task (or subtasks) to the original problem.

(iii) Write a complete algorithm using pseudocode representation for each of the module you have specified above.

[40/100]

(c) (i) Write the function's prototype in C for the following descriptions:

A function **test** receiving three parameters, the first is an array of **float**, the second parameter is an **integer**, and the third parameter is a pointer to **double**. Function **test** will return a data of type pointer to integer.

(ii) Explain the main purpose of function's prototype in C environment.

[20/100]

(d) Based on the variables declared below, what is the outputs produced by the following C statements (note that each question is independent from one another):

```
int x = 0, y = 10, z = 20;
```

(i)
```
z++;
x += y++ + z;
printf (" x = %d,  y = %d,  z = %d\n", x, y, z);
```

(ii)
```
if (!x && y <= z)
    printf (" x = %d,  y = %d,  z = %d\n", ++x, y*10, z-10);
else
    printf (" x = %d,  y = %d,  z = %d\n", x+10, y--, ++z);
```

(iii)
```
x =  y * z / 5 * 4;
printf (" x = %d\n", x);
```

[10/100]

2. (a) Given the following array declaration, write a function (for each of the following cases), which will increment all elements of array **x** by 2.

```
int x[5] = {10, 30, 50, 70}
```

(i) Using call by value.

(ii) Using call by reference.

(iii) Why, sometimes it is effecient to call a function by reference? Justify your answer by refering to example use in 2 (a) (i).

[25/100]

(b) Your task is to test a function **ComputeGrade** which computes the exam grade (A-D) corresponding to a given mark (in the range 0 to 100 inclusive). Marks of 70 or more are awarded A, from 50 to 69 inclusive a B, from 30 to 49 inclusive C, and below 30 a D. The function specification is as follows:

Name:		ComputeGrade
Parameters:		
IN	Mark	The raw mark (an integer)
OUT	Grade	The corresponding grade
OUT	OK	set to true if the mark is valid (in the range 0 - 100)

Description: If **Mark** is valid, **OK** returns true and **Grade** returns the grade corresponding to **Mark**. If the mark is not valid, **OK** returns false and **Grade** is undefined.

Side effects: none.

(i) Write a function for the above specification.

(ii) Identify the few test cases and hence test data to test the function. Prepare the test plan in table format which include the test data, purpose (test cases) and the expected results.

[30/100]

(c) Given the following variables declarations, what output(s) each of the following statements produce? Note that statements in each questions are not related.

```
char name[] = "CSI502 - Principles of Programming";
char *nPtr, ch;
char addr[80];
```

(i)
```
nPtr = name;
printf ("%s", nPtr);
```

(ii)
```
nPtr = name; ch = *nPtr;
printf ("%c", ch);
```

(iii)
```
nPtr = name + 9; strcpy (addr, nPtr);
printf ("%s", addr);
```

(iv)
```
nPtr = name + 6; *nPtr = '\0';
printf ("%s",name);
```

(v)
```
nPtr = name;
for (; (ch = *nPtr++)!= '\0';)
printf ("%c",ch);
```

[10/100]

(d) Write a loop that will calculate the sum of every third integer, beginning with i = 2, (i.e. Calculate the sum 2+5+8+11+ ...) for all values of **i** less than 100. Write the loop in three different ways.

(i) Using a **while** loop statement.

(ii) Using a **do - while** loop statement.

(iii) Using a **for** loop statement.

[20/100]

(e) (i) What outputs does the following program produce?

```
#include <stdio.h>

void a (int);
void b (void);
void c (void);

int  y = 5;

main() {
    int   y = 10;

    a (y);
    b ();
    printf ("main --> %d\n",y);
}

void a (int y) {
    int x = 10;

    printf ("a --> %d\n",x+y);
    y += 20;
    c();
}

void b () {
    y *= 2;
    printf ("b --> %d\n",y);
    c();
}

void c () {
    static int y  = 30;

    printf ("c --> %d\n",y);
    y += 20;
}
```

(ii) What **#include** and **#define** preprocessors do? Give example to support your answer.

[15/100]

3. (a) Define a structure that contains the following members:

- a string of size 80 **characters** called **name.**
- an **integer** quantity called **id.**
- a self-referential pointer called **link.**

[15/100]

(b) Use the **typedef** statement to define a new-type for structure defined in 3 (a) above to a type called **patient.**

[5/100]

(c) Declare the following variables:

- a pointer variable to data type **patient** called **p_start.**
- a variable of type **patient** called **patient1.**

[10/100]

(d) Using the variables declared in 3 (c), write C statements to accomplish the following tasks:

(i) Initialise the variable **patient1** with the following values:

- **name** member of **patient1** with string "Bailey"
- **id** member of **patient1** with 1007.
- **link** member of **patient1** with NULL.

(ii) Initialise variable **p_start** with similar value you assign to variable **patient1** in 3 (d)(i) above.

[10/100]

(e) Write a complete C program which will do the following tasks:

The program will create a link list structure which store the details of patients. The program will prompt the user to enter patient's particulars. These data will then be stored in a patient data structure, before it is inserted to a link list. The insertion must be done in the ascending order of the patient's name. The program should also give a choice to the user to either terminate or to continue the insertion process. Duplicate record of patient is allowed.

Note: Use the data structure you defined in 3 (a) above.

[40/100]

(f) Describe the differences between the following concepts:

(i) Stack and Queue.

(ii) Static structured data and dynamic structured data.

[20/100]

4. (a) Write a C program which will open and read the contents of a random-access file called **story.txt** sequentially. The program will then print the text it reads to the standard output. The file **story.txt** consists of a series of short stories text, where each of the short story has a fix size of 80 characters.

[30/100]

(b) List the advantages and disadvantages of random-access file as compared to the sequential access file.

[20/100]

(c) List down several factors which will cause function **fopen** to return NULL value during the process of opening file.

[10/100]

(d) (i) Describe the following concepts of object-orientation:

- Polymorphism
- Information Hiding
- Inheritence

[20/100]

(ii) What is the difference between class and instance?

[10/100]

(iii) How code reusability can be realised in object-oriented languages?

[10/100]

- oooOooo -